

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Absolvování individuální odborné praxe
Individual Professional Practise in the Company

2010

Lukáš Kohut

Zadání bakalářské práce

Student:

Lukáš Kohut

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Absolvování individuální odborné praxe
Individual Professional Practise in the Company

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: ATLAS consulting spol. s r.o.
2. Struktura závěrečné zprávy:
 - a. Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
 - b. Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
 - c. Zvolený postup řešení zadaných úkolů.
 - d. Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
 - e. Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
 - f. Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Doc. Mgr. Jiří Dvorský, Ph.D.**

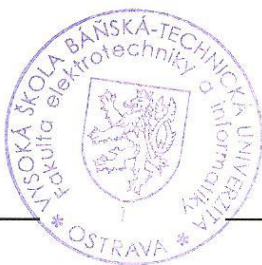
Konzultant bakalářské práce: Ing. Zdeněk Růžička

Datum zadání: 20.11.2009

Datum odevzdání: 07.05.2010



doc. Dr.Ing. Eduard Sojka
vedoucí katedry



prof. Ing. Ivo Vondrák, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 7. května 2010

.....
podpis

Poděkování

Tímto bych chtěl poděkovat vedení firmy Atlas Consulting spol. s r. o. za poskytnutí příležitosti k vykonání odborné praxe, odbornému konzultantovi Zdeňku Růžičkovi za jeho trpělivost a vstřícnost při vedení odborné praxe a také Ladislavu Paliderovi za pomoc při řešení provedených úkolů.

Abstrakt

Tato bakalářská práce se zabývá stručným popisem průběhu absolvování odborné praxe ve firmě Atlas Consulting spol. s r. o. Je zde souhrnně sepsáno řešení zadaných úkolů a seznámení s modulárním systémem Obchodní kancelář, na kterém byly úkoly prováděny. V jednotlivých kapitolách věnované úkolům jsou obsaženy informace o použitých technologiích a postupu řešení. V závěru jsou shrnuty dosažené výsledky v průběhu praxe a její celkové zhodnocení.

Klíčová slova

Odborná praxe, Obchodní kancelář, Delphi, BDE, Interbase, Virtual TreeView, Pascal

Abstract

This bachelor labour makes brief overview focusing at my time when i was working in the company Atlas Conculting spol. Ltd. Is here collectively drafted resolution assignments and an introduction to modular systems the Business office, where they were performed. In individual chapters devoted to the tasks contained information about the technology and process solutions. In conclusion, the results are summarized in the practice and its overall assessment.

Keywords

Professional practice, Business office, Delphi, BDE, Interbase, Virtual treeview, Pascal

Seznam použitých zkratek

BDE – Borland Database Engine

XML - Extensible Markup Language

HTML - HyperText Markup Language

SQL – Structured Query Language

Obsah

| | | |
|-------|---|----|
| 1. | Úvod | 1 |
| 2. | Popis firmy a pracovního zařazení | 2 |
| 3. | Seznámení s produktem Obchodní kancelář | 3 |
| 4. | Zadané úkoly | 4 |
| 4.1 | Úkol č.1 - Převod přístupu k databázi z BDE na Interbase | 4 |
| 4.1.1 | Zadání | 4 |
| 4.1.2 | Postup řešení | 4 |
| 4.1.3 | Zhodnocení | 6 |
| 4.2 | Úkol č.2 - Aktualizace komponenty Virtual TreeView | 6 |
| 4.2.1 | Zadání | 6 |
| 4.2.2 | Postup řešení | 6 |
| 4.2.3 | Zhodnocení | 7 |
| 4.3 | Úkol č.3 - Převod nápovědy z formátu HLP na CHM | 7 |
| 4.3.1 | Zadání | 7 |
| 4.3.2 | Postup řešení | 7 |
| 4.3.3 | Zhodnocení | 8 |
| 4.4 | Úkol č.4 - Zajištění multijazyčnosti | 8 |
| 4.4.1 | Zadání | 8 |
| 4.4.2 | Postup řešení | 8 |
| 4.4.3 | Zhodnocení | 9 |
| 4.5 | Ladění systému | 9 |
| 5. | Znalosti a dovednosti scházející v průběhu praxe | 10 |
| 6. | Teoretické a praktické znalosti získané během studia uplatněné studentem v průběhu odborné praxe | 11 |
| 7. | Závěr | 12 |
| 8. | Literatura | 13 |

1. Úvod

V bakalářské práci se Vám pokusím shrnout mou působnost ve firmě Atlas consulting, za účelem vykonání bakalářské praxe. Seznámím Vás s úkoly, kterými jsem byl pověřen. Úkoly budu chtít shrnout v jednotlivých kapitolách, ve kterých se budu věnovat popisu použité technologie, přístupu k řešení a popřípadě řešení problémům, které se vyskytly v rámci vykonaného řešení. Provedené úkoly jsou rozděleny na čtyři podkategorie, ve kterých se zabývám převodu přístupu k databázi s BDE na Inter-Base, aktualizaci komponenty Virtual TreeView, převodu nápovědy z HLP na CHM a zajištění multi-jazyčnosti.

2. Popis firmy a pracovního zařazení

Praxi jsem vykonával ve firmě Atlas Consulting spol. s r. o. sídlící v Ostravě. Společnost se zabývá vývojem a distribucí právních a ekonomických informačních systémů pro malé, střední a velké firmy.

Během praxe jsem byl zařazen do týmu, zabývající se vývojem modulárního systému Obchodní kancelář, na pozici „Programátor Delphi – externí pracovník“, kde jsem se podílel jak na vývoji nových požadavků, tak na obměně stávajícího řešení.

3. Seznámení s produktem Obchodní kancelář

Obchodní kancelář je modulární systém aplikací pro řízení obchodní agendy, vztahů se zákazníky a navazujících činností. Řeší nejen oblast vztahů mezi firmou a jejími klienty, ale i vztahy firmy samotné.

Moduly Obchodní kanceláře:

- **Aktivity a komunikace se zákazníky**
je určena ke sledování a analýze kontaktu s klienty. Kontaktem se rozumí jakýkoliv vstupní nebo výstupní přenos dat, informací, názorů, písemností a zásilek.
- **Řízení prodeje**
je prostředek k organizaci a řízení práce obchodních zástupců a pracujících v přímém kontaktu s koncovými zákazníky. Umožňuje evidovat veškeré obchodní aktivity, sledovat průběh kontaktů a modelovat nabídky a objednávky s využitím ceníku firemních produktů.
- **Zákaznická podpora a řešení případů**
modul je určen pro systematické řešení problémů ke kterým může dojít při obchodní činnosti – reklamace, stížnosti, požadavky na technickou nebo metodickou podporu, servisní činnost. Systém umožňuje sledovat celý postup vývoje, případu od jeho vzniku až k uzavření a potvrzení výsledku.
- **Hromadná korespondence**
modul je určen pro hromadné kontaktování vybraných klientů. Využit lze elektronické pošty, případně exportovat kontaktní údaje ke zpracování softwarem třetích stran.
- **Avíza**
slouží pro předávání zpráv mezi uživateli v rámci jedné či více databází a pro automatické informování uživatelů o stavu v databázi pomocí alarmu a kontrolních zpráv.

Produkt je vyvíjen vývojovým nástrojem Delphi, který umožňuje jednoduše navrhovat a efektivně vytvářet aplikace pod operačním systémem Windows. Prostředí Delphi je založeno na jazyce Object Pascal, což je objektová nadstavba nad jazykem Turbo Pascal. Při vývoji Obchodní kanceláře bylo použito vývojové prostředí Delphi 5.

4. Zadané úkoly

4.1. Úkol č.1 - Převod přístupu k databázi z BDE na Interbase

4.1.1. Zadání

Hlavním úkolem bylo provést změnu přístupu k databázi z původní použité technologie BDE na novější technologii Interbase. Starší technologie způsobovala potíže v souvislosti s provozem Obchodní kanceláře na nových operačních systémech jako je Windows Vista, a to tehdy kdy uživatel používal dvě aplikace současně. Také tato technologie před pár lety ztratila podporu ze strany Delphi. Z tohoto důvodu bylo navrženo přejít na novější a podporovanější technologii Interbase, která byla uvolněna jako open source. Díky tomu došlo k podpoře open source komunity a spojení s databázovou platformou Firebird. Změna spočívala ve výměně použitých komponent BDE za komponenty Interbase a změny v kódu, kde se muselo nahradit řešení pomocí BDE přístupu za novější.

4.1.2. Postup řešení

Z důvodu velkého množství zdrojového kódu a rozsáhlosti projektu, bylo velice důležité zvolit nejvhodnější cestu k provedení úkolu. Zvolená strategie odrážela postupy, které byly použity v jiných projektech.

V prvním kroku došlo k nahrazení původních komponent ve všech použitých formulářích. Náhrada komponent se prováděla pouze u komponent, patřících do skupiny nevizuálních komponent, to zajistilo, že změny se neprojeví na vzhledu jednotlivých formulářů. Při vkládání nových komponent na formuláře se nesmělo zapomenout na nastavení ve vlastnostech, kde muselo dojít k nastavení propojení jednotlivých komponent.

Použité komponenty:

| Původní BDE | Nové Interbase |
|--------------------|-----------------------|
| Database | IBDatabase |
| Query | IBQuery |
| UpdateSql | IBUpdateSql |
| Table | IBTable |
| - | IBTransaction |

Tabulka č.1: Přehled použitých komponent

Přehled použitých komponent ukazuje na podobnosti mezi jednotlivými platformami. Je to i z důvodu zpětné kompatibility. Podstatná změna nastává u komponenty IBTransaction, která se u původní platformy vůbec nevyskytuje. V platformě BDE nelze asociovat transakce ke komponentám, nelze realizovat práci s více transakcemi nad jedním připojením. Nová platforma Interbase naopak umožňuje asociaci transakce k jednotlivým komponentám, to umožňuje používání více transakcí nad jedním připojením a větší kontrolu nad jednotlivými operacemi.

Dalším časově náročným krokem převodu platformy BDE na Interbase, bylo procházení jednotlivých zdrojových kódů a vyhledávání míst, kde byla potřeba provést korekce použitých starých částí kódu, které používaly starou platformu. Zdrojové kódy obsahovaly dynamické vytváření komponent, které byla potřeba zaměnit za komponenty platformy Interbase a také operace prováděny nad starými komponentami se lišily u staré a nové platformy. Mezi upravované operace patřilo připojení na databázi, propojení databáze s komponentami a transakcemi.

Příklady změn v kódu:

| Původní BDE | Nové Interbase |
|-----------------------------|---|
| Datasase.InTransaction | IBDatabase.DefaultTransaction.InTransaction |
| Query.Database.DatabaseName | IBQuery.Database |
| Database.Commit | IBDatabase.DefaultTransaction.Commit |
| Database.Rollback | IBDatabase.DefaultTransaction.Rollback |

Tabulka č.2: Příklady změn v kódu

Poslední krok převodu spočíval v odinstalování platformy BDE. Tím jsem zajistil úplnou nezávislost kódu na staré platformě. Po odinstalování jsem mohl přistoupit na ladění aplikace a začít provádět opravy na zapomenutých částech kódu, které nebyly upraveny na novou platformu. Tento krok byl prováděn opakovaně, než jsem docílil spuštění aplikace.

4.1.3. Zhodnocení

Testování funkčnosti systému ukázalo, že striktnější pravidla nové platformy způsobují pády aplikace. Tyto pády zapříčinila komponenta TIBQuery, která byla na mnohých místech otevřená, a po vykonání operace nebylo provedeno korektní uzavření. Problém byl vyřešen kontrolou všech komponent TIBQuery ve zdrojových kódech a dodatečným dopsáním uzavření komponenty. Další problémy systému vznikly při přepisu kódů, kdy docházelo v několika případech k překlepům. Tyto problémy byly opraveny individuálně. Úkol splnil všechny zadané podmínky a přinesl praktické zkušenosti s programováním databázových platforem v Delphi.

4.2. Úkol č.2 - Aktualizace komponenty Virtual TreeView

4.2.1. Zadání

Úkol byl věnován aktualizaci komponenty Virtual TreeView, která byla použita ve verzi 1.31. Důvodem aktualizace byly problémy při použití scrollbaru v komponentě, kdy nedocházelo k přesnému překreslování zobrazovaných dat. Tento problém měla řešit verze 4.0.15. Tato moderní komponenta slouží k zobrazování strukturovaných dat. Pracuje na architektuře stromu, kdy vytváříme jednotlivé uzly a strukturujeme je pomocí propojení s jinými uzly. Výsledné data máme možnost zobrazit jako tabulku nebo strom. Komponenta byla použita v systému kvůli své rychlosti zobrazování dat a nízké zátěži operační paměti.

4.2.2. Postup řešení

Z důvodu velice špatné dokumentace k původní verzi, nemohlo dojít k detailnímu porovnání obou verzí pomocí dokumentace. Řešení teda spočívalo ve vyhledávání použitého řešení ve staré verzi a porovnání s dokumentací nové verze. Analýza ukázala rozdílné parametry u definicí a volání jednotlivých metod.

Řešení úkolu spočívalo v odstranění staré verze komponenty a nahrání nové verze. Vyhledání použitých komponent ve zdrojovém kódu a následné sekvenční korekce jednotlivých metod podle nové dokumentace.

4.2.3. Zhodnocení

Kvůli častému výskytu komponenty v systému se jednalo o časově náročný úkol. Aktualizace komponenty opravila všechny problémy způsobené starší verzí.

4.3. Úkol č.3 - Převod nápovědy z formátu HLP na CHM

4.3.1. Zadání

Mezi další zadané úkoly bylo provést převod nápovědy z formátu HLP na formát CHM. Provedení úkolu také spočívalo v zajištění zobrazení nápovědy v novém formátu. Důvodem změny nápovědy byla nekompatibilita s novými operačními systémy Windows Vista a Seven.

4.3.2. Postup řešení

Díky kompatibilitě původní nápovědy s vývojovým nástrojem Delphi 5, byla práce s nápovědou usnadněná. Stačilo stanovit ve vlastnostech projektu, který soubor s nápovědou se vztahuje k aplikaci. Poté pokud máme správně namapované kontextové řetězce na identifikační čísla, stačí nastavit vlastnost *HelpContext* k jednotlivým komponentám. Následně stačí vyvolat nápovědu pomocí tlačítka F1 nad komponentou. O zobrazení nápovědy se Delphi staralo programově samo. Další možností bylo vyvolání nápovědy programově příkazem *Application.HelpContext(1)* s číselným parametrem odkazujícím na kontextový řetězec. Naopak formát CHM kompatibilní s vývojovým nástrojem Delphi 5 nebyl. Z tohoto důvodu musela být naimplementována vlastní třída pro volání nápovědy formátu CHM.

Prvním krokem bylo vytvoření souboru nápovědy formátu CHM pomocí kompletních zdrojových textů ve formátu HTML stránek. Všechny texty byly převzaty z původního souboru HLP. Poté byla potřeba vytvoření hierarchického stromu obsahující jména témat a přiřazení jednotlivých HTML stránek pod témata. Nakonec stačila provést kompilace pomocí programu překladače helpů, jehož výstupem je konečný soubor CHM.

Po vytvoření souboru nápovědy jsem naimplementoval vlastní třídu zajišťující zobrazení nápovědy. Třída zajišťovala zobrazení nápovědy pomocí procedury *ShowHtmlHelp(HelpChm:string)*, kde parametr *HelpChm* obsahoval název CHM souboru nebo zobrazení pomocí přetížené procedury *ShowHtmlHelp(HelpChm:string; Topic:string)*, kde parametr *Topic* obsahoval název požadovaného tématu.

V rámci úkolu převodu nápovědy z formátu HLP na CHM, bylo požadováno také vytvoření kontextové nápovědy. Tento druh nápovědy byl použit ve dvou z modulů systému. Požadavek

byl sjednotit jednotlivé moduly, aby obsahovali stejnou kontextovou nápovědu. Kontextová nápověda měla být zobrazovaná v pravé části obrazovky ve formátu HTML. Uživatel si mohl nastavit, zda nápovědu chce viditelnou nebo jak široká má být.

Pro vytvoření kontextové nápovědy jsem použil komponentu TWebBroser, která zajišťovala zobrazení nápovědy v HTML formátu. Pomocí vhodně použitých panelů a separátorů jsem zajistil uživatelsky nastavitelnou šířku nápovědy. Pro zobrazování a skrývání nápovědy byla použita viditelnost jednotlivých komponent.

4.3.1. Zhodnocení

Tento úkol přinesl velmi bohaté teoretické znalosti a praktické zkušenosti z oblasti tvorby uživatelských nápověd. Úkol byl vypracován podle všech požadavků a splnil představu vedení. Při provádění úkolu nedošlo k žádným komplikacím.

4.4. Úkol č.4 - Zajištění multijazyčnosti

4.4.1. Zadání

Poslední prováděný úkol, měl zajistit připravenost aplikace na překlad do více jazyků. Úkol měl být proveden pomocí třídy Langs, která sloužila pro práci s jazykovou sadou.

4.4.2. Postup řešení

Pro práci s jazykovou sadou existovala třída Langs, která načítala dynamicky z xml souboru textové vlastnosti komponent a konstant použitých v jednotlivých aplikacích. Identifikace textové vlastnosti komponenty se prováděla pomocí elementu „PROPERTY“ a atributů:

- název aplikace
- název formuláře
- název komponenty
- typ textové vlastnosti

Příklad zápisu vlastnosti formuláře:

```
<PROPERTY project="OK" form="TfrmMain" component="acLogin" name="Caption"
value="Přihlásit"/>
```

Identifikace konstanty se prováděla pomocí elementu „CONST“ a atributů:

- název aplikace
- unikátní identifikátor

Příklad zápisu konstanty:

```
<CONST project="OK" name="MT_MERGESOURCECOMPANY"
    value="Zdrojová společnost: "/>
```

Pokud se použitá konstanta objevovala ve více aplikacích, měl jsem možnost použití globální proměnné, která se zapisovala bez identifikátoru názvu aplikace. To zajistilo, že stejná konstanta nebyla duplicitní ve více aplikacích.

Můj úkol spočíval připravit systém na multijazyčnost. To znamenalo kontrolu všech komponent použitých v aplikacích, zda se nachází v jazykovém xml souboru a obsahuje všechny zadané textové vlastnosti. Dále kontrola všech řádků zdrojového kódu, zda jsou použity jazykové řetězce pomocí volání funkce ze třídy Langs. Kontrola se prováděla sekvenčně jak na straně formuláře, tak na straně zdrojových kódů. V případě, že při kontrole byla nalezena komponenta, která nebyla přiřazena v jazykovém souboru, musela být dodatečně přidána. Pokud byl nalezen jazykový řetězec ve zdrojovém kódu, který neobsahoval volání funkce ze třídy Langs, byla potřeba vložit nový záznam do jazykového souboru pod unikátní identifikátor a poté nahradit jazykový řetězec za volání funkce, která byla volána s identifikátorem jako atributem.

4.4.3. Zhodnocení

Vypracování tohoto úkolu zajistilo použití systému ve více jazykových mutacích pomocí xml souboru. Úkol přinesl cenné informace o práci se soubory xml ve spojení s multijazyčností.

4.5. Ladění systému

V rámci absolvování odborné praxe jsem prováděl ladění chyb vzniklých při vývoji systému. Ladění jsem prováděl po celou dobu provádění individuální odborné praxe. Mým úkolem bylo analyzovat kroky vedoucí k vyvolání chyby, popřípadě vymyslet postup řešení. Zjištěné poznatky byly konzultovány s vedením. Jestliže se jednalo o složitý zákrok zasahující do primární funkčnosti systému, došlo k delegování chyby na zkušenějšího kolegu, v jiném případě jsem provedl opravu.

5. Znalosti a dovednosti scházející v průběhu praxe

V průběhu praxe scházely znalosti z oblasti pokročilejšího programování v Delphi. Tyto nedostatky byly nahrazeny díky samostudia během provádění individuální odborné praxe. Nastudované znalosti byly použity pro splnění jednotlivých zadaných úkolů. Chybějící znalosti byly čerpány z odborné literatury a internetových stránek věnované programování v Delphi.

6. Teoretické a praktické znalosti získané během studia uplatněné studentem v průběhu odborné praxe

Díky získaných teoretických a praktických znalostí během studia z oblasti programování a práce s databázemi se přistupovalo k jednotlivým úkolům odborněji. Tím bylo ulehčeno vypracování jednotlivých úkolů. Studium usnadnilo rychlejší orientaci v přístupu k databázi, práci s transakcemi a vytváření SQL dotazů. I když se jednalo o programování v Delphi, došlo k rychlému přechodu a nalezení podobností s jinými programovacími jazyky probíraných v rámci studia.

7. Závěr

V průběhu praxe byly úspěšně splněny všechny zadání úkolů. Řešení jednotlivých úkolů bylo konzultováno s vedením a také schváleno. Vypracované změny provedené v informačním systému opravily všechny problémy způsobené s kompatibilitou s novými operačními systémy. Během praxe jsem získal neocenitelné zkušenosti s velice rozsáhlými informačními systémy a práce s jejich databázemi. Možnost absolvovat individuální odbornou praxi umožnilo porovnat rozdíly v řešení ve firemním a univerzitním prostředí. Díky řešení jednotlivých úkolů samostatně jsem zjistil, jak cenné jsou všeobecné znalosti získané během studia.

8. Literatura

KADLEC, V. Delphi – Hotová řešení. Brno: Computer Press, a.s. 2005. ISBN 80-251-0017-0

SVOBODA, L. Přecházíme na Delphi 5. Brno: Computer Press, a.s. 2000. ISBN 80-7226-288-2

TEIXERA, T. a PACHECO, X. Borland Delphi průvodce vývojáře. Mobil Media, a.s. 2002. ISBN 80-86593-10-X

ŽIVĚ. Internet. [Online], Url: <http://www.zive.cz/clanky/umime-to-s-delphi--1-dil/sc-3-a-30959/default.aspx>, 2.5.2010